Action Recognition on Motion Capture Data Using a Dynemes and Forward Differences Representation

Ioannis Kapsouras, Nikos Nikolaidis

Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki 541 24, GREECE

Abstract

In this paper we introduce a novel method for action/movement recognition in motion capture data. The joints orientation angles and the forward differences of these angles in different temporal scales are used to represent a motion capture sequence. Initially K-means is applied on training data to discover the most representative patterns on orientation angles and their forward differences. A novel K-means variant that takes into account the periodic nature of angular data is applied on the former. Each frame is then assigned to one or more of these patterns and histograms that describe the frequency of occurrence of these patterns for each movement are constructed. Nearest neighbour and SVM classification are used for action recognition on the test data. The effectiveness and robustness of this method is shown through extensive experimental results on four standard databases of motion capture data and various experimental setups.

Keywords: Dynemes, Forward Differences, Action Recognition, Bag of Words, Motion Capture Data

Email address: `[jkapsouras,nikolaid]@aiia.csd.auth.gr` (Ioannis Kapsouras, Nikos Nikolaidis)

Preprint submitted to Journal of Visual Communication and Image Representation April 9, 2014

1. Introduction

Motion capture (mocap) data provide a representation of the complex spatio-temporal structure of human motion. During a traditional motion capture session, the locations of characteristic parts on the human body such as joints or the joint angles are recorded over time, using appropriate tracking devices [1]. Different tracking technologies (magnetic, ultrasonic, inertial, optical, mechanical) are in use today. Moreover, motion capture can nowadays be performed with the use of low-cost 3D capturing devices such as the Microsoft Kinect sensor. Motion capture data, usually in the form of joint angles, are used in computer games and movies to animate a hierarchical structure (skeleton) representing a human [2], where the nodes model the joints of the skeleton and the arcs the segments (links). Some examples of mocap data are shown in Fig. 1.

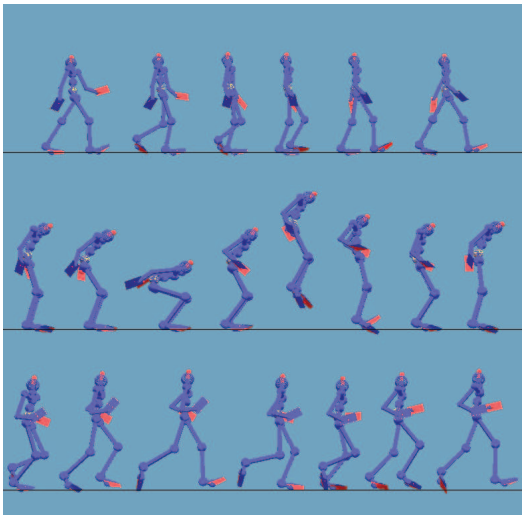


Figure 1: Walk, hop and run movement sequences from the HDM05 database [3].

A node's degrees of freedom depend on the allowable rotations and translations of the corresponding joint in the skeleton. Usually all joints have 3 rotational degrees of freedom (with respect to each of the three axes) whereas the root node has also 3 translation parameters. The angle values of a certain frame form the n-dimensional pose or posture vector. An example skeleton is shown in Fig. 2.

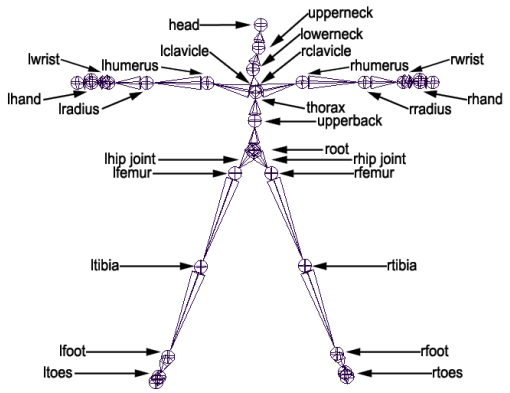


Figure 2: A skeleton representation of human body.

Action recognition is an active research topic that deals with the process of labelling a motion sequence with respect to the depicted motions. Technically, an action is a sequence generated by a human subject during the performance of a task. Action recognition is usually performed on video data and has many applications including human computer interaction, video surveillance, multimedia annotation and retrieval, etc.

The corpus of research dealing with action recognition on motion capture data is more recent. Action recognition in such data has many applications

including automatic annotation of motion capture data for archival, indexing, retrieval and asset management in games or animated movies production, robotics, etc. It should be noted that capture devices such as Microsoft Kinect, along with the corresponding software are able to generate multimodal action data consisting of video, depth and motion capture sequences. Thus mocap-based action recognition on such data can be directly integrated with algorithms that perform action recognition on video data. A few multimodal databases such as MHAD [4] and MSR Action3D [5] databases have been created.

An action recognition method usually comprises of two basic steps. The first step, namely feature extraction, deals with the transformation of the input data into an appropriate representation that increases the separability of the data that belong to different classes. The next step involves the training of models from the extracted features that are able to classify different action classes. This is a challenging task, because the same movements can be performed in a different way by different subjects. Moreover, the same person can perform a movement with different speed, style, etc.

This paper presents a movement recognition method that operates on motion capture data i.e. skeleton animation data. The method involves the posture vector at each frame namely the vector of rotation angles in the selected skeleton joints. Forward differences at different temporal scales for each joint angle are also computed in order to capture the dynamics of each joint. K-means is applied separately on the postures space and on the forward differences of the training data to discover characteristic patterns. Then each posture and each forward difference vector is mapped to the pat-

terns and histograms with the frequencies of patterns appearances for each sequence are formed. Classification is implemented using two different approaches: a nearest neighbour approach that utilizes histogram intersection or Kullback-Leibler divergence (KL) as distance measure or by using support vector machines (SVM).

The use of forward differences of postures vectors in different temporal scales as a new representation for motion capture data is one of the novel contributions of the proposed method. An additional contribution is the use of a modified K-means algorithm that can handle angular data such as the joint angles involved in mocap. The need for such a modification arises from the fact that angular data are periodic and their natural representation is on the unit circle. Thus, the notions of distance and mean value for angular data are different from those for data on the line. The proposed method follows the bag of words (BoW) framework with two differences/novelties: K-means is applied on each set of features separately, resulting in multiple histograms that describe a sequence, and histograms are implemented using a voting scheme. The method does not require segmentation of motion capture sequences into "atomic" movements such as steps.

The proposed method has been shown, through extensive experimental evaluation, to outperform most of the state-of-the-art methods in four well known datasets.

The remaining of this paper is organized as follows. In Section 2, we present a review of previous work on this topic. In Section 3, the proposed method is described in detail. Experimental performance evaluation of the method and comparison with other approaches is presented in Section 4.

Conclusions follow in Section 5.

2. Previous Work

A great amount of research has been performed on activity recognition from video data. Surveys of activity recognition approaches on such data can be found in [6], [7] and [8]. However, motion capture technology became widely available only during the last years. Hence the body of research for movement recognition on mocap data is not as extensive as for video data. Li et al. [9] propose a method for action recognition and classification of motion capture data. Their method uses singular value decomposition (SVD) to extract feature vectors from motion data. SVM classifiers are used to segment and recognize motion streams. SVM classification applied on the vector of 3D locations of characteristic points on the human body is used by Wang et al. in [10] for human movement recognition. Kadu et al. [11] adopt the tree-structured vector quantization (TSVQ) method to represent human poses by codewords and approximate the dynamics of mocap sequences by a codeword sequence. For the classification, the authors use a spatial domain approach based on the histogram of codewords and a spatiotemporal domain approach via codeword sequence matching. An algorithm for sequence alignment and activity recognition, called IsoCCA, is described in [12]. IsoCCA extends the canonical correlation analysis (CCA) algorithm, by means of introducing a number of alternative monotonicity constraints. The activity classification task performed in this paper is based on a 1-NN classifier, that uses the alignment cost between sequences as distance metric. The method yields improved classification rates in comparison

to other alignment algorithms, such as Canonical Time Wrapping (CTW), Dynamic Time Wrapping (DTW), Hungarian and CCA. In [13] the authors introduce a method for real-time classification of dance gestures in skeletal animation. An angular skeleton representation that maps the motion data to a smaller set of features is used. The full torso is fitted with a single reference frame. This frame is used to parametrize the orientation estimates of both the first-degree limb joints (joints adjacent to torso) and second-degree limb joints (tips of the skeleton extremities such as the hands and the feet). Then a cascaded correlation-based maximum-likelihood multivariate classifier is used to build a statistical model for each gesture class. The trained classifier compares the input data with the gesture model of each class and outputs a maximum-likelihood score. An input gesture is finally compared with a prototype one using a distance metric that involves dynamic time-warping. In [14], Lv et al. present a method for movement recognition where each movement is represented as a spatio-temporal template consisting of a set of channels with weights. The channels correspond to the 3D joints trajectories and the weights are learned according to the Neyman-Pearson criterion. Movements are recognized by comparing them with the templates. In [15], Deng et al. propose a method for human motion recognition. First the method partitions a human model in five parts, namely, torso, left upper limb, right upper limb, left lower limb and right lower limb and K-means is applied separately to each of these partitions. Then several trials from each K-means class are used to train a generalized model to represent that class. For isolated motion recognition the authors propose a voting scheme that can be used with common dynamic programming techniques and they

also present a new penalty-based similarity measure for DTW. For continuous motion recognition, five body partition index maps are constructed and applied. Concepts from the theory of chaotic systems are used by the framework proposed by Ali et al. in [16] to model and analyze nonlinear dynamics of human actions. The authors use the trajectories of reference body joints to create time series by considering each data dimension separately. Mutual information and false nearest neighbourhood algorithms are used to embed each time series in a phase space of an appropriate dimension. Phase space invariants are then used to represent the dynamical and metric structure of the phase space. The invariants from all time series are subsequently used to generate a global feature vector of an action. These feature vectors are used as input in a K-nearest neighbor classifier. Ofli et al. [17] proposed a new representation for skeletal action recognition. The authors use either a fixed number of segments or a fixed temporal window, to segment an action sequence. Subsequently, they find in these segments the most informative joints and to represent the sequence with the use of these joints. Nearest neighbour and SVM are used for classification. A hierarchical discriminative approach is used in [18] by Han et al. for human action recognition. At first, the authors use hierarchical latent variable space analysis to represent the human motion in a hierarchical manifold space. Mutual invariant features from each manifold subspace are extracted by the use of conditional random fields. Finally, an SVM classifier is used for the classification.

Action recognition in skeleton animation data became more popular with the release of markerless capture devices such as Microsoft Kinect. The Kinect sensor not only records depth video but also provides, through al-

gorithms included in the software that accompanies the sensor, information for the joints positions of the tracked skeletons providing the possibility to combine the video action recognition with motion capture action recognition. Li et al. in [5] proposed a method for action recognition in depth video data without the use of the corresponding tracked skeleton. They construct an action graph to encode human actions and propose a bag of 3D points approach to characterize a set of salient postures that correspond to the nodes in the action graph. They also propose a projection method to sample the 3D points from the depth maps. The 3D skeletal joint locations, extracted from Kinect depth maps, were used from Xia et al. in [19] to perform action recognition. The authors represent the human postures by histograms of 3D joint location and use Linear Discriminant Analysis (LDA), clustering and vector quantization to encode action sequences. Finally, classification is performed with the use of hidden Markov models. The 3D joint positions were also used by Yang and Tian in [20] to perform action recognition. The authors proposed a new type of features to represent a sequence of skeletal motion. They model the dynamics of individual joints by computing pairwise differences of joints in the spatial and temporal domain. Moreover, they use principal component analysis to reduce redundancy and noise and to obtain EigenJoints. Finally, they use a non-parametric Naïve-Bayes-Nearest-Neighbor (NBNN) classifier for action classification. The authors in [20] compute differences between all pairs of joints x_i, x_j in the current frame and also between the current and the previous or the initial frame, whereas in the proposed method the difference between a joint's values in the current and subsequent frames, i.e., temporal differences, are evaluated. The method

proposed in [21] by Wang et al. also operates on 3D joints positions. The authors propose a new set of features called local occupancy pattern and a new temporal patterns representation called Fourier Temporal Pyramid. They represent an action as a linear combination of acionlets, where an actionlet is a particular conjunction of the features for a subset of the joints. SVM is used for the classification. In [22], the authors propose a method for 3D human action recognition. The authors use motion trajectories to represent each human action. Moreover, the motion trajectories are projected in a Riemmanian shape space. Finally, recognition is performed using a k nearest neighbour approach that takes into account the geodesic distances in the Riemmanian space. Gowayyed et al. in [23] proposed histograms of oriented displacements (HOD) for action recognition. The displacement of each joint votes with its length in a histogram of oriented angles. Each 3D trajectory is represented by the HOD of its three 2D projection. In order to capture temporal information, the authors apply a temporal pyramid, where they compute the trajectories as a whole, halves and then quarters. Classification is performed using a linear SVM on the histograms. Barnachon et al. in [24] proposed a method that uses histograms of 3D motion capture data for action recognition. Histograms of action poses are computed to represent an action and then Dynamic Time Wrapping is used to compare and recognize actions. A method for recognizing actions using depth maps is proposed by Yang et al. in [25]. The authors use Histograms of Oriented Gradients to represent an action. The HOGs are extracted from Depth Motion Maps (DMM) where DMM are computed by stacking motion energy of depth maps projected onto three orthogonal planes. The authors in [25] use

the Depth Motion Maps, hence they use video data instead of skeletal data. A linear SVM is used for the classification. Oreifej and Liu in [26] proposed a method for action recognition from depth sequences. The authors represent a sequence by forming a histogram of the surface normal orientation in time, depth and space. Moreover, they use a novel discriminative density measure to refine the quantization. Finally, they use SVM for classification.

Several methods in the literature use the bag of words framework to perform action classification. Raptis et al. in [27] construct dictionaries of action primitives in order to perform action classification. They use multi-dimensional time series to represent the deformation of the body during an action. More specifically, they track the positions of limb endpoints (arms, legs and head) and create action dictionaries from the temporal scale, mean and shape of primitive motion trajectories. A bag of words approach is used on the action primitives for the classification. Their method works both on video and motion capture data. In [28], Wang et al. estimate human joints positions from videos in order to perform action recognition. The authors group the estimated joints into five parts and then represent each action by computing sets of co-occuring spatial configurations of body parts and sets of co-occuring pose sequences of body parts. The authors use a bag of words approach with the extracted features for classification. Ohn-Bar and Trivedi in [29] proposed two set of features for action recognition on both depth video data and skeletal data. As for the depth video data, the authors use a modified Histogram of Oriented Gradients. They compute histograms at each frame in box regions around each joint and then they re-apply the algorithm on these histograms to capture temporal dynamics. Regarding the

skeletal data, they use affinities within sequences of joint angles. Features extracted from depth video and skeletal data are used to represent an action sequence in a bag of words approach. The proposed method differs from the bag of words methods presented above since it uses a variant, where the bag of words is applied in each feature set separately resulting to different sets of histograms, instead of concatenating the features before the application of the bag of words framework. Moreover, as already stated in Section 1, we proposed a variant that uses a voting scheme for computing the histograms. More details are presented in Section 3.

Action segmentation and continuous action recognition can be also performed on motion capture data. Barbič et al. [30] proposed three methods for automatic action segmentation on a motion capture sequence. In the first one, a boundary between consecutive actions is recognized using an indication of intrinsic dimensionality from Principal Component Analysis while the second method extracts segments using Probabilistic PCA. A Gaussian mixture model representation in used in the third one. Raptis et al. in [31] model an action as a linear time-invariant dynamical model of relatively small order, driven by a sparse bounded input signal. The authors use the temporal statistics of the input sequences for recognizing and detecting transitions between actions. A method for temporal segmentation of human motion in motion capture data is proposed by Zhou et al. in [32]. The authors propose Aligned Cluster Analysis (ACA) as an extension of k-means clustering. A variable number of features in the cluster centers and a dynamic time warping kernel for temporal invariance achievement are the two extensions to the classical k-means algorithm introduced in ACA.

3. Method Description

In the proposed method each movement is represented by two types of features: the posture vectors and the forward differences vectors.

Posture Vectors: A motion capture sequence can be represented as a sequence of posture vector $\mathbf{x}_i,\ i = 1, \ldots, N$ where N is the number of frames of the sequence. Each posture vector carries information for the rotation angles in the selected skeleton joints

$$\mathbf{x}_i = [\theta_{i1}, \theta_{i2}, \ldots, \theta_{in}] \tag{1}$$

where n is the number of rotation angles that form the posture vector.

Forward Differences: A motion capture sequence can also be represented by vectors of joints angles forward differences. Forward differences estimate the first derivative of a signal and thus, when applied on joint angles, carry information for the average angular velocities of the skeleton joints within a temporal interval. More specifically, forward differences in terms of a motion capture sequence can be defined as:

$$\boldsymbol{\upsilon}_i^t = \Delta_t[\mathbf{x}] = \mathbf{x}_{i+t} - \mathbf{x}_i \tag{2}$$

where $\mathbf{x}_i, \mathbf{x}_{i+t}$ are the posture vectors in frames i and $i+t$ respectively. In the proposed method, the forward differences of the joints angles are computed in different temporal scales and more specifically for $t = 1$, $t = 5$ and $t = 10$ in order to capture the dynamics of the joints of a skeleton. It should be noted that since in (2) the values in posture vectors $\mathbf{x}_{i+t}$ and $\mathbf{x}_i$ are angular values, the evaluation of the difference

of two vectors involves finding for each pair of corresponding elements their angular distance, which will be defined in Section 3.2 (equation (4)).

Summarizing, two types of features, forming 4 groups of vectors are used to represent a motion capture sequence: posture vectors and forward differences vectors in three different temporal scales. Thus a sequence is represented by four sets of feature vectors $\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3, \mathbf{M}_4$:

$$\begin{aligned} \mathbf{M}_1 &= \{\mathbf{x}_1, \ldots, \mathbf{x}_N\} \\ \mathbf{M}_2 &= \{\boldsymbol{v}_1^1, \ldots, \boldsymbol{v}_{N-1}^1\} \\ \mathbf{M}_3 &= \{\boldsymbol{v}_1^5, \ldots, \boldsymbol{v}_{N-5}^5\} \\ \mathbf{M}_4 &= \{\boldsymbol{v}_1^{10}, \ldots, \boldsymbol{v}_{N-10}^{10}\} \end{aligned} \tag{3}$$

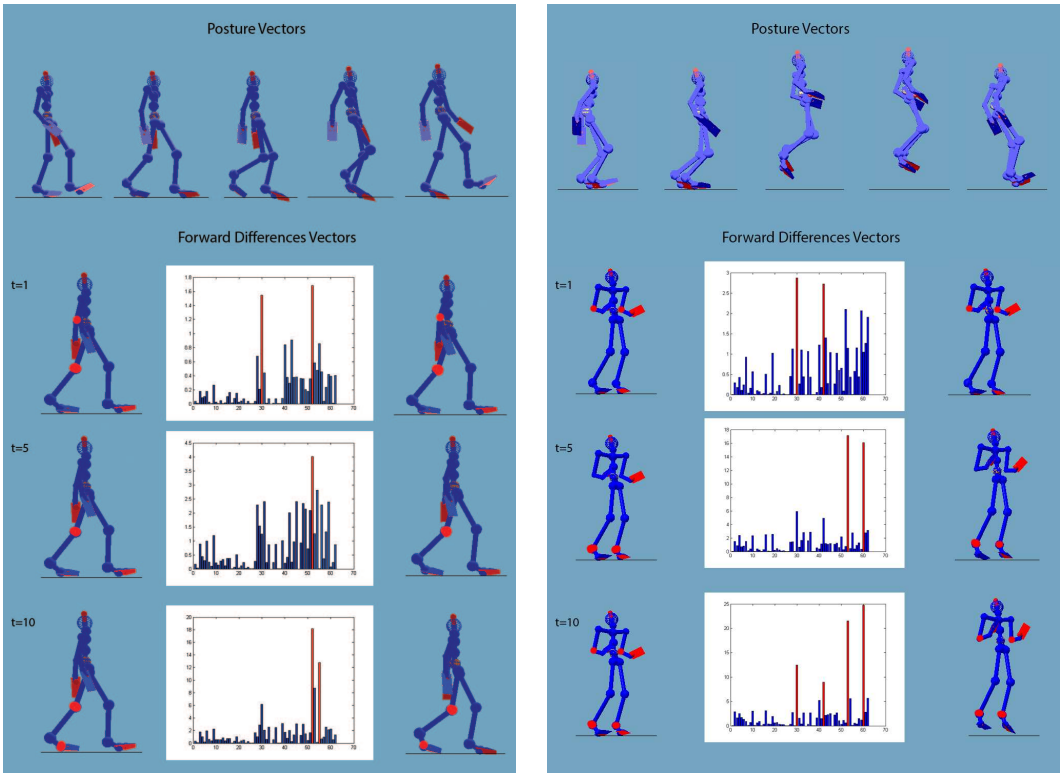
Features (postures and forward differences) are shown in Fig. 3.

The basic building blocks of the method are presented in the following subsections.

3.1. Feature extraction

The first step of the algorithm concerns the quantization of the vector spaces and the extraction, through clustering, of four codebooks consisting of characteristic words.

Indeed, in order to recognize K different movement classes, we cluster feature vectors in each feature space (the postures space and the forward differences spaces in different temporal scales) into C clusters. The clusters are identified by unsupervised clustering, using the K-means algorithm. K-means is applied four times. First K-means is applied on all the posture



(a) (b)

Figure 3: a) Features of a walk sequence b) Features of a hop sequence. The postures are shown as skeletons whereas forward differences in the different temporal scales are shown as bar plots The joints with the highest forward differences are drawn with red color.

vectors $\mathbf{x}_{ij}, i = 1, \ldots, N_j, j = 1, \ldots, L$ of all movement sequences in the training set where N_j is the number of frames of the j-th movement sequence and L the number of the training sequences. In other words K-means is applied upon sets $\mathbf{M}_1$ of all training sequences. Since elements of posture vectors are joint angles, a novel K-means variant modified to work on angular data is applied (see Subsection 3.2). For each cluster created by the angular K-means algorithm, the centroid $\mathbf{v}_c, c = 1, \ldots, C$ is computed as the circular

mean (Subsection 3.2) of all postures in this cluster. This centroid represents one dyneme. Due to the averaging procedure, dynemes don't correspond to postures from the training set but rather on "average", characteristic postures. Example dynemes are shown in Fig. 4. Moreover, K-means is applied on the forward differences vectors $\boldsymbol{v}_{ij}^t, i = 1, \ldots, N_j - t, j = 1, \ldots, L$, each time for a different temporal scale $t = 1, 5, 10$ where $N_j - t$ the number of forward differences vectors in the j sequence and t the temporal scale. This results to $3 \cdot C$ centroids, $\mathbf{z}_c^t, c = 1, \ldots, C$. The number C of clusters is selected empirically and depends on the number of movements K that are to be recognized, the different ways a movement can be performed by different people, the different body types, etc. However, as shown in Section 4 the performance of the algorithm is not significantly affected by C, when C is selected within a fairly large range of values. It should be noted that due to the way forward differences are evaluated (equations (2) and (4)) they take values in the range $[0, \pi]$ and not $[0, 2\pi]$ and thus both the classical K-means algorithm and its angular variant (Subsection 3.2) generate the same results.

3.2. Angular K-means

Motion capture data, i.e. posture vectors $\mathbf{x}_i$, describe rotation angles at the joints. As already mentioned, due to the periodic nature of angular data neither the Euclidean distance nor the mean value estimator for data on the line can be used in such data. For example, an angle of 0 radians is the same as an angle of 2π radians but their Euclidean distance wouldn't be zero but 2π. Furthermore, the average of two angles $5°$ and $355°$ is $0°$ and not $180°$ as the classical average operator would entail. Two different measures, namely the distance between two angles and the circular mean [33]

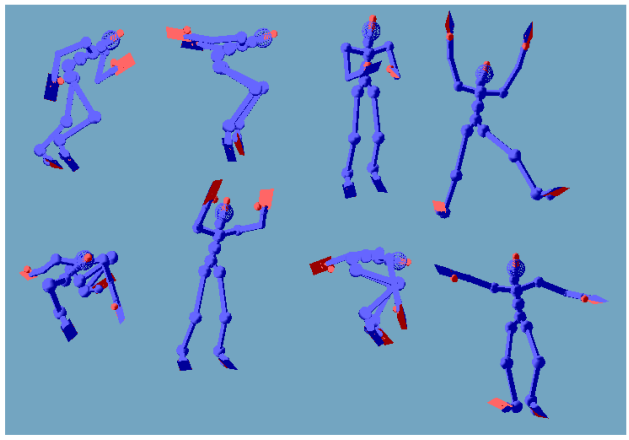


Figure 4: Examples of dynemes extracted by the clustering algorithm.

can be used instead. The distance between two angles θ_i, θ_j is the smallest arc between the two points that are defined by these angles on the unit circle:

$$arc(\theta_i, \theta_j) = \pi - \mid \pi - \mid \theta_i - \theta_j \mid\mid \tag{4}$$

The *circular mean* or *sample mean direction* $\bar{x_0}$ of k angular observations $\theta_1, \cdots, \theta_k$ represented by sample points $\mathbf{P}_1, \cdots, \mathbf{P}_k$ on a unit circle centred at point $\mathbf{O}$ is the direction of the mean resultant vector $\mathbf{R}$ of the unit vectors $\mathbf{OP}_1, \cdots, \mathbf{OP}_k$. Its value is given by:

$$\bar{x_0} = \arctan\left(\frac{\bar{S}}{\bar{C}}\right),\ \bar{C} = \frac{1}{k}\sum_{i=1}^{k}\cos\theta_i,\ \bar{S} = \frac{1}{k}\sum_{i=1}^{k}\sin\theta_i \tag{5}$$

Since the proposed algorithm applies the K-means algorithm on angle data (posture vectors), a modified angular version was constructed by replacing the classical mean and Euclidean distance by the above quantities.

3.3. Evaluation of the bags of words

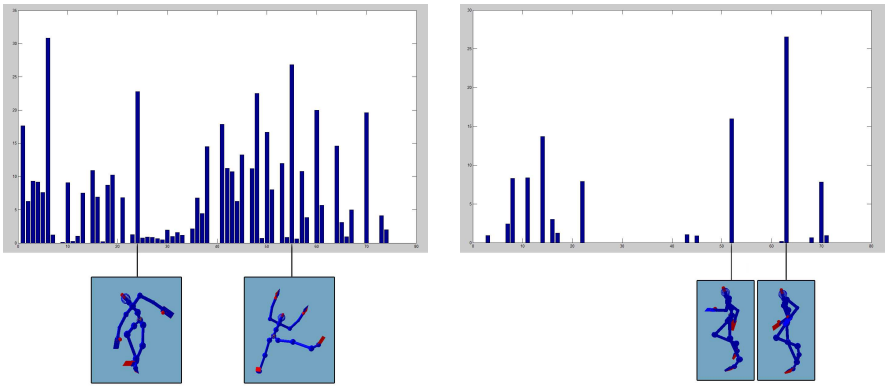
Two variants were considered for this step. In the first one (variant 1) we map each posture vector $\mathbf{x}_{ij}$ and forward differences vector $\boldsymbol{\upsilon}_{ij}^{t}$ in the training set to a cluster center. In more detail, the angular K-means algorithm will assign each vector to a class $c = 1, \ldots, C$. Based on the clustering results we map each vector to its assigned class centroid. Thus each movement sequence is represented in terms of cluster centroids (words). More specifically, each frame is represented by four cluster centroids, i.e. those that correspond to the posture vector and the three forward differences vectors of this frame. Frames at the end of each sequence for which forward differences cannot be evaluated are represented by less than four centroids.

Next, we form for each sequence four histograms $\mathbf{s}_j^{\mathbf{x}}, \mathbf{s}_j^{\boldsymbol{\upsilon}^1}, \mathbf{s}_j^{\boldsymbol{\upsilon}^5}, \mathbf{s}_j^{\boldsymbol{\upsilon}^{10}}$ by calculating the frequency of appearance of every cluster center for the four types of features. The histogram of the j-th sequence is a C-dimensional vector $\mathbf{q}_j = [q_{ij}], j = 1, \ldots, L, i = 1, \ldots, C$:

$$q_{ij} = \frac{n_i}{N_j} \tag{6}$$

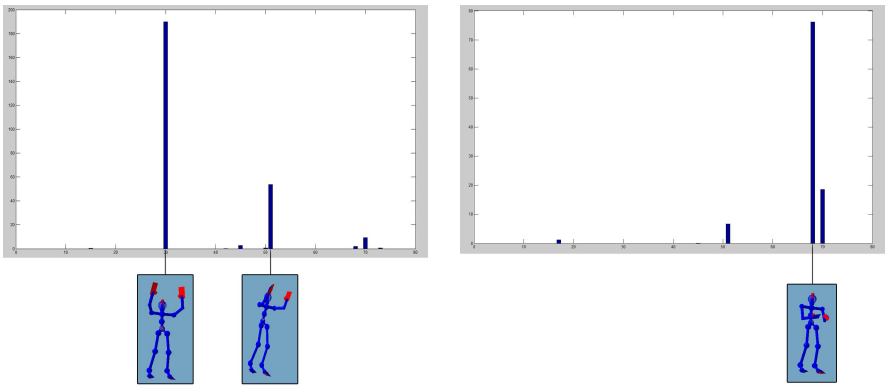
where n_i is the number of occurrences of the i-th center within the sequence and N_j the number of feature vectors of the sequence. Example histograms of posture vectors are shown in Figs 5, 6 whereas some histograms of forward differences vectors are shown in Fig. 7. Finally, the four histograms are concatenated to form the final vector having $4 \cdot C$ elements that characterizes the movement:

$$\mathbf{s}_j = \left[\mathbf{s}_j^{\mathbf{x}}, \mathbf{s}_j^{\boldsymbol{\upsilon}^1}, s_j^{\boldsymbol{\upsilon}^5}, \mathbf{s}_j^{\boldsymbol{\upsilon}^{10}}\right] \tag{7}$$



(a) (b)

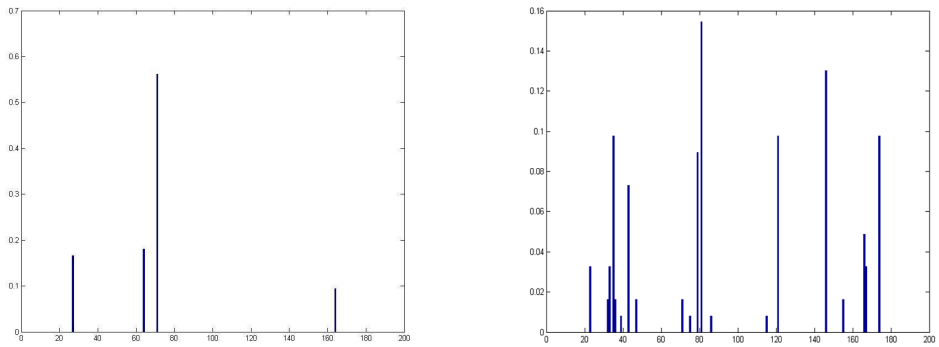
Figure 5: Dynemes histograms of cartwheel (a) and run on place (b) movements generated by the proposed method, along with some dynemes that correspond to the most prominent peaks of the histograms.



(a) (b)

Figure 6: Dynemes histograms of clap above head (a) and clap (b) movements generated by the proposed method, along with some dynemes that correspond to the most prominent peaks of the histograms.

In a second variant of the method (variant 2) instead of assigning each feature vector to a single cluster center, we use a voting approach. At first



(a) (b)

Figure 7: Forward differences ($t = 1$) histogram of (a) deposit floor movement and (b) clap above head movement.

we evaluate the similarities of each feature vector with each cluster center:

$$d_k = sim(\mathbf{v}_k, \mathbf{x}) = exp(-(\frac{\sum_{i=1}^{n}(arc(v_{ki} - x_i))}{0.5 * \max_k(\sum_{i=1}^{n}(arc(v_{ki} - x_i)))})^2) \quad (8)$$

where d_k is the similarity of cluster center $\mathbf{v}_k$ and feature vector $\mathbf{x}$. The similarity measure used in (8) was proposed in [34]. The distance values within the exponential are normalized by half the maximum distance of the feature vector from the cluster centers. By using this equation, the similarities between a feature vector and the closest cluster centers are high, while the similarities with more distant cluster centers are low. A more common approach is to use the sum of all distances as a denominator, but with this approach more cluster centers achieve high similarities. Then we form for each feature vector, a vector of ordered similarities: $D = [d_{(1)}, \ldots, d_{(C)}]$, where C is the number of clusters. Finally, we form a C dimensional histogram for each type of features by adding to the bin that corresponds to a cluster center the similarity of the feature vector with this center, starting

from the most similar until the sum of the R largest similarities surpasses the 5% of the sum of all similarities. In other words, R is found as the value that satisfies the following inequalities:

$$\frac{\sum_{k=1}^{R-1} d_{(k)}}{\sum_{k=1}^{C} d_{(k)}} < 0.05 < \frac{\sum_{k=1}^{R} d_{(k)}}{\sum_{k=1}^{C} d_{(k)}} \tag{9}$$

In the special case where $d_{(1)} > 0.05 \sum_{k=1}^{C} d_{(k)}$, R is set to 1. An example of this procedure in two dimensions is shown in Fig. 8. By choosing (8) as a similarity measure, the number R of the similarities to be added to the corresponding bins of the histogram is kept relatively small. The four histograms that result from the procedure are then concatenated into a single vector $\mathbf{s}_j$ that characterized the movement.

Unlike the first variant of the method, that assigns each feature vector of a frame to a single cluster center, in the second variant each feature vector of a frame is represented by its similarities to the closest cluster centers, which is a richer representation. Thus, while in the first variant a value equal to one is added to the histogram bin that corresponds to the cluster center where the frame/feature vector has been assigned, in the second variant the similarities of the feature vector with the closest cluster centers are added to the respective bins. The intuition behind (9), which essentially selects the number of the closest centers for which the corresponding similarities will be added, is that a feature vector/frame is well represented by its closest centers whereas similarities with distant centers provide little information and can be regarded as noise. Experiments showed that large values of R (including $R = C$) decrease the performance of the method.

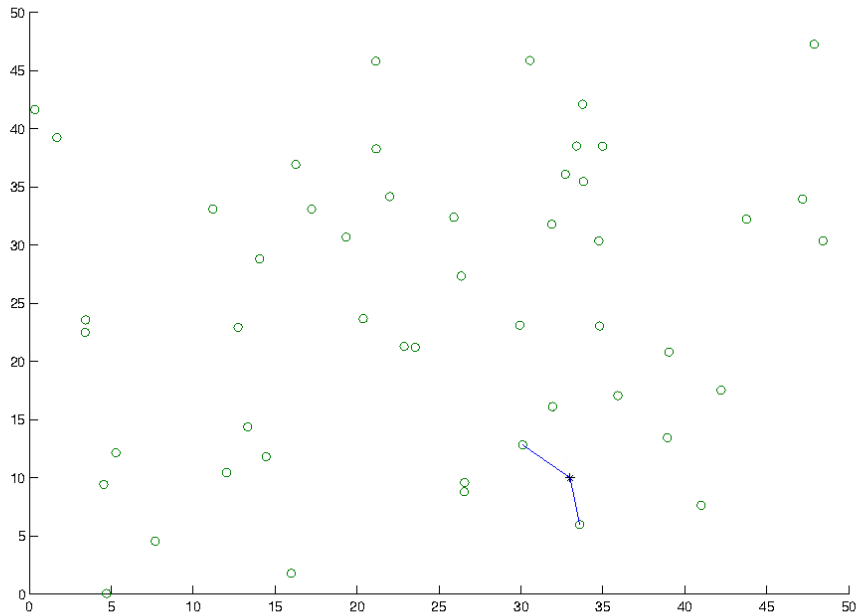


Figure 8: The star represents a feature vector and the circles are the cluster centers. The star is connected with the 2 most similar centers. The sum of the similarities to the 2 most similar centers surpasses the 5% of the sum of the similarities to all the cluster centers, while the similarity of the first most similar center does not surpass the above percentage. Hence according to (9), $R = 2$ and only the first 2 similarities will be added in the corresponding bins of the histogram.

3.4. Classification

To classify an unknown motion capture sequence to one of the movements that the algorithm has been trained to recognize the following procedure is used.

For the first variant of the method, we first use the posture vectors cluster centers $\mathbf{v}_C$ evaluated during the training stage to map each posture vector $\mathbf{x}$

of the test sequence into the nearest center:

$$k = \arg\min_{c \in [1 \ldots C]} (\sum_{i=1}^{n} arc(v_{ci} - x_i)) \tag{10}$$

Then we map each forward difference vector $\boldsymbol{\upsilon}^t$, for each temporal scale t of the testing sequence into the nearest center:

$$k = \arg\min_{c \in [1 \ldots C]} (\sum_{i=1}^{n} |z_{ci}^t - \upsilon_i^t|) \tag{11}$$

Once all vectors have been mapped to cluster centers we calculate four histograms $\mathbf{q}_{test}^{\mathbf{x}}, \mathbf{q}_{test}^{\boldsymbol{\upsilon}^1}, \mathbf{q}_{test}^{\boldsymbol{\upsilon}^5}, \mathbf{q}_{test}^{\boldsymbol{\upsilon}^{10}}$ for the testing sequence using (6) or (9) and the four histograms are concatenated to form the final $\mathbf{q}_{test} = \left\{ \mathbf{q}_{test}^{\mathbf{x}}, \mathbf{q}_{test}^{\boldsymbol{\upsilon}^1}, \mathbf{q}_{test}^{\boldsymbol{\upsilon}^5}, \mathbf{q}_{test}^{\boldsymbol{\upsilon}^{10}} \right\}$ histogram. $\mathbf{q}_{test}$ characterizes the sequence and is used for the classification/recognition. For the second variant, the $\mathbf{q}_{test}$ vector is similarly created by using the voting approach described in the previous subsection. Two classifiers were used, a Nearest Neighbour (NN) classifier and a Support Vector Machine (SVM) classifier. Details are provided below.

3.4.1. Nearest neighbour classification

In this case, histogram intersection (12) and a symmetric version of Kullback-Leibler divergence (13) are used to measure the similarity of $\mathbf{q}_{test}$ with all labelled movement sequences of the training set:

$$HI(\mathbf{s}_j, \mathbf{q}_{test}) = \sum_{i=1}^{4 \cdot C} min\{s_{j,i}, q_{test,i}\} \tag{12}$$

$$KL(\mathbf{s}_j, \mathbf{q}_{test}) = \frac{1}{2}(\sum_{i=1}^{4 \cdot C} s_{j,i} \ln \frac{s_{j,i}}{q_{test,i}} + \sum_{i=1}^{4 \cdot C} q_{test,i} \ln \frac{q_{test,i}}{s_{j,i}}) \tag{13}$$

where $s_{j,i}$ and $q_{test,i}$ are the values of the i-th element for the vector $\mathbf{s}_j$ of the j-th training sequence and the test sequence vector $\mathbf{q}_{test}$ respectively. Since both $\mathbf{s}_j$ and $\mathbf{q}_{test}$ are vectors that result from the concatenation of histograms, each of the equations (12) and (13) essentially evaluates the sum of histogram intersection and Kullback-Leibler divergence values for the four histograms that make up vectors $\mathbf{s}_j$ and $\mathbf{q}_{test}$.

The test sequence is subsequently assigned to the movement label of the training set movement sequence whose similarity with the test sequence is maximum.

3.4.2. SVM classification

In this case, an SVM classifier is trained using the vectors $\mathbf{s}_j$ of the labeled sequences of the training set. The trained SVM is then used to classify the vector $\mathbf{q}_{test}$ of an unknown sequence.

Frequently used kernel functions include the polynomial kernel, the Radial Basis Function (RBF) kernel and the χ^2 kernel as proposed in [35]:

$$K(\mathbf{x}_j, \mathbf{x}_k) = exp(-\frac{1}{2A}\sum_{i=1}^{4\cdot C}\frac{(x_{j,i} - x_{k,i})^2}{x_{j,i} + x_{k,i}}) \tag{14}$$

where A is the mean value of distances between all training samples. Experiments conducted in this paper showed that the RBF kernel and the χ^2 had the best results for the K-class SVM that was implemented.

4. Experimental Results

The proposed method has been tested on various datasets from four databases namely *HDM05, Multimodal Human Action Database (MHAD),*

MSR Action3D (MSR) and *Futurelight* using various validation approaches. All datasets but the MSR Action3D dataset are motion capture datasets. These datasets contain rotation angles at the corresponding joints for many body parts (see Figure 2). Since some of these body parts, for example the thumb or the toes, bear no significant information for action recognition, in the *HDM05* dataset, we have retained only information for the following body parts: lower back, upper back, thorax, right humerus, right radius, left humerus, left radius, right femur, right tibia, right foot, left femur, left tibia, left foot. In a similar manner, information for the following parts has been retained in the *MHAD* and *Futurelight* datasets: hips, spine, neck left and right shoulder, left and right arm, left and right arm roll, left and right fore arm, left and right up leg, left and right up leg roll, left and right leg and left and right foot. In addition, the global translation and rotation information (i.e. the rotation and translation of the root node) were not considered by the algorithm. The MSR Action3D dataset contains sequences obtained with the Kinect device. The skeleton of this dataset contains only 20 joints, thus no joints were excluded in this dataset. In all experiments the involved histograms were normalized so that their elements sum to one. For the SVM classifier with the RBF kernel the γ and soft margin parameters were set to 0.75 and 1000 respectively. The SVM classifier with the χ^2 kernel was trained with values of the soft margin parameter in the range $2^{-20}, 2^{-19}, \ldots, 2^{19}, 2^{20}$ and the best results are presented. The evaluation procedure, experimental results and comparison with other methods for each database are presented in different subsections. The number of clusters that led to the best performance is mentioned in separate columns in each results table.

4.1. Datasets from HDM05 database

The HDM05 database [3] consists of various movements performed by five subjects in the form of amc files. Two different datasets, subsets of HDM05, were used for the experiments. The first dataset (HDM05 Dataset 1), proposed by Deng et al. in [15] includes 10 actions namely *clap above hands (ClapAbove), deposit floor (DepositFloor), jumping jack (JumpingJacks), front kick with right foot (FKickR), front punch with right hand (FPunchR), run on place (RunP), squat (Squat), staircase up (StaircaseUp), stand up sit floor (StandSit) and walk left (WalkL)*. The three fold validation setup, proposed in [15], was used. In more detail, the sequences of three subjects were selected to form the training set and the the sequences of the other two to form the testing set in every validation cycle. The distribution of training and testing sequences in the 3 validation cycles are shown in Table 1.

Table 1: Distribution of training and testing subjects in the three validation cycles proposed in [15].

Validation cycles	**Subjects for training**	**Subject for testing**
Cycle 1	1,2,3	4,5
Cycle 2	3,4,5	1,2
Cycle 3	1,2,5	3,4

The correct classification rates for all the variants of the proposed method and a comparison with the results presented in [15] are shown in Table 2. Variants 1 and 2 refer to the two alternative approaches described in Section 3.

As can be seen in this table the proposed method outperforms the method

Table 2: Correct Classification rates and comparison of classification performance in the experimental setup proposed in [15] on the HDM05 Dataset 1.

	10 action classes	**Number of Clusters**
SVM-χ^2 (variant 1/2)	**100 / 98.88**	100 / 197
SVM-RBF (variant 1/2)	94.38 / 96.07	188 / 186
NN-Intersection (variant 1/2)	94.94 / 96.07	183 / 205
NN-KL (variant 1/2)	97.19 / 94.94	187 / 194
GM + SW [15]	94.96	-

proposed in [15] in most of the cases. SVM with χ^2 kernel achieved the highest classification rate, regardless the type of the variant.

The second subset used by Ofli et al. in [17] (HDM05 Dataset 2) includes 16 actions namely *deposit floor (DepositFloor), elbow to knee (ElbowKnee), grab high (GrabHigh), hop both legs (HopBoth), jog (Jog), kick forwards (KickFor), lie down floor (LieFloor), rotate both arms backward (RotateBArmsB), sneak (Sneak), squat (Squat), throw basketball (ThrowBasket), jump (Jump), jumping jacks (JumpJacks), throw (Throw), sit down (SitDown) and stand up (StandUp)*. The experimental setup proposed in [17] was used. In more detail the sequences of 3 subjects were used to form the training set (216 sequences) and the sequences from the other 2 subjects were used to form the test set (177 action sequences). The correct classification rates for all the variants and a comparison with [17] are shown in Table 3.

As can be seen in Table 3 the proposed method outperforms the method proposed in [17] when SVM with χ^2 kernel is used as a classifier. The con-

Table 3: Correct Classification rates and comparison of classification performance in the experimental setup proposed in [17] on the HDM05 Dataset 2.

	16 action classes	**Number of Clusters**
SVM-χ^2 (variant 1/2)	**95.48 / 93.22**	500 / 253
SVM-RBF (variant 1/2)	88.70 / 89.83	217 / 202
NN-Intersection (variant 1/2)	88.70 / 85.88	217 / 205
NN-KL (variant 1/2)	85.31 / 82.49	211 / 216
SMIJ [17]	91.53	-

fusion matrix for the best variant (SVM-χ^2 with variant 1) is shown in Fig. 9. As can be seen, the proposed method recognize with lower classification rates (below 85%) only the SitDown action, which is confused with the similar StandUp action and the Throw action, which is confused with the similar ThrowBasket action.

4.2. MHAD database

The Berkeley Multimodal Human Action Database (MHAD) [4] contains 11 actions performed by 12 subjects in the form of bvh files. Each subject performs each action 5 times. The dataset contains the following actions: *jump (JumpPlace), jumping jacks (JumpJacks), bend (Bend), punch (Punch), wave one hand (WaveOne), wave two hands (WaveTwo), clap (Clap), throw (ThrowBall), sit down (Sit), stand up (Stand) and sit down/stand up (SitStand).* The experimental setup proposed in [17] was used, where sequences of 7 subjects were used to form the training set (384 action sequences) and the sequences from the remaining 5 subjects were used to form the testing set (275 action sequences). The correct classification rates

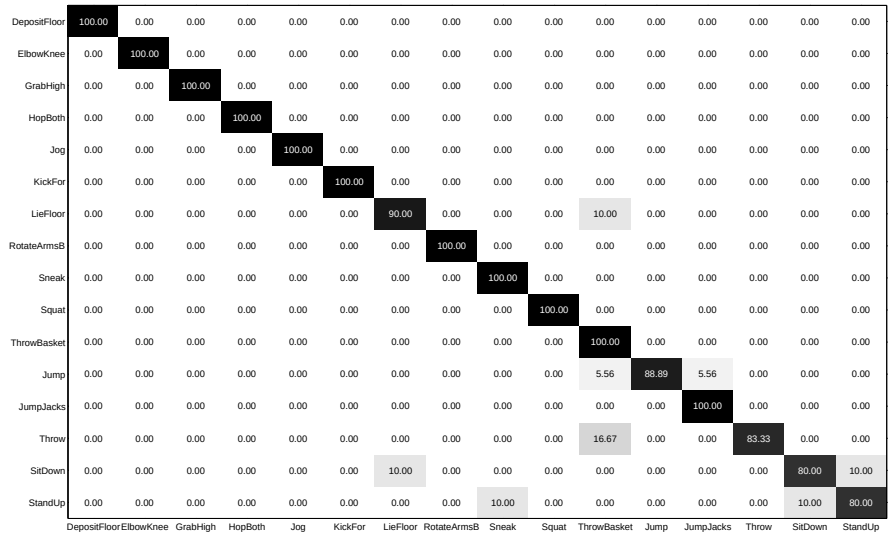


Figure 9: Confusion matrix (16 classes) for SVM-χ^2 with variant 1 classification with $C = 500$; 95.48% overall correct classification rate. HDM05 Dataset 2.

for all the variants and a comparison with [17] is shown in Table 4. The pro-

Table 4: Correct Classification rates and comparison of classification performance in the experimental setup proposed in [17] on the MHAD database.

	11 classes	**Number of Clusters**
SVM-χ^2 (variant 1/2)	**98.18** / **97.82**	186 / 214
SVM-RBF (variant 1/2)	93.82 / 95.64	208 / 195
NN-Intersection (variant 1/2)	88.36 / 84.36	215 / 218
NN-KL (variant 1/2)	86.18 / 76.72	100 / 185
SMIJ [17]	95.37	-

posed method outperforms the method proposed in [17] when SVM with χ^2

	JumpPlace	JumpJacks	Bend	Punch	WaveTwo	WaveOne	Clap	ThrowBall	SitStand	Sit	Stand
JumpPlace	96.00	0.00	0.00	0.00	0.00	0.00	4.00	0.00	0.00	0.00	0.00
JumpJacks	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Bend	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Punch	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
WaveTwo	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00
WaveOne	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00
Clap	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00
ThrowBall	12.00	0.00	0.00	0.00	4.00	0.00	0.00	84.00	0.00	0.00	0.00
SitStand	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00
Sit	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00
Stand	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00

Figure 10: Confusion matrix (11 classes) for SVM-χ^2 with variant 1 classification with $C = 186$; 98.18% overall correct classification rate. MHAD Dataset.

kernel is used a a classifier and when SVM-RBF is used with variant 2 of the method. The confusion matrix for the best variant (SVM-χ^2 with variant 1) for this dataset is shown in Fig. 10. In this experimental setup all actions but two are perfectly recognized.

4.3. Datasets from MSR Action3D database

The MSR Action3D database [5] contains sequences of skeleton animation data obtained from a Microsoft Kinect sensor and therefore are more noisy than the sequences obtained from "traditional" (and more expensive) motion captures systems. The database consists of 10 subject performing 20

actions with 2 or 3 repetitions of each action. Two different datasets were used for the experiments. In the first dataset, a subset of 8 subjects and 17 actions (MSR1), namely *high arm wave (HighArmW), horizontal arm wave (HorizArmW), hammer (Hammer), hand catch (HandCatch), forward punch (FPunch), high throw (HighThrow), draw x (DrawX), draw tick (DrawTick), draw circle (DrawCircle), hand clap (Clap), two hand wave (TwoHandW), side-boxing (Sidebox), forward kick (FKick), side kick (SKick), jogging (Jog), tennis swing (TSwing)* and *tennis serve TServe*, were used for the experimental evaluation of the method as proposed in [17]. The experimental setup proposed in [17] was used. According to this setup sequences of 5 subjects were used to form the training set (226 action sequences) and the sequences from the remaining 3 subjects were used to form the testing set (153 action sequences). It should be noted that the MSR Action3D database provides the positions of the joints instead of the corresponding orientations. Thus, in order to evaluate the proposed method in this specific dataset, we modified the method so to operate on the joint positions. More specifically, the angular K-means was replaced by the classical K-means and the angular differences were replaced with the classical differences in the Euclidean space. It should be also noted that PCA was applied to the positions of the joints to decorrelate the data. The correct classification rates for all the variants and a comparison with the [17] are shown in Table 5. It should be noted that the authors in [17] converted the joint positions to joint angles.

As can be seen in this table the proposed method by far outperforms the method proposed in [17] regardless the variant used in classification for both types of variants. The confusion matrix for the best variant (SVM-χ^2 with

Table 5: Correct Classification rates and comparison of classification performance in the experimental setup proposed in [17] on the MSR Action3D dataset.

	17 classes	**Number of Clusters**
SVM-χ^2 (variant 1/2)	**87.74** / **90.97**	200 / 206
SVM-RBF (variant 1/2)	83.87 / 85.16	214 / 197
NN-Intersection (variant 1/2)	81.93 / 78.06	219 / 195
NN-KL (variant 1/2)	81.93 / 82.58	219 / 208
SMIJ [17]	33.99	-

variant 2) for the MSR dataset proposed in [17] is shown in Fig. 11. 13 out of 17 classes are recognized with 100% classification rate in this challenging dataset. To ensure fair comparison with [17], in another experiment, the joint angles evaluated by the joint positions as in [17] were used to assess the performance of the method. The joint angles were computed between any two adjacent links in the skeleton hierarchy, considering the two links as vectors and the joint angle as the angle between these vectors. However, since the links/vectors are in 3D, using only their angle results in an incomplete representation. Essentially such a representation is the axis-angle representation, without the rotation axis. The result of this experiment are shown in Table 6. As can be seen the proposed method again outperforms the method proposed in [17] even when the joint angles are used as input. The degraded performance of the proposed method in this case is most probably due to the incomplete representation mentioned above.

The second dataset consists of the entire database (MSR2) and besides the 17 actions mentioned above, includes *bend (Bend), golf swing (Golf) and*

Table 6: Correct Classification rates and comparison of classification performance in the experimental setup proposed in [17] on the MSR Action3D dataset using joint angles evaluated by converting joint positions.

	17 classes	**Number of Clusters**
SVM-χ^2 (variant 1/2)	**54.90** / **52.94**	194 / 204
SVM-RBF (variant 1/2)	47.06 / 50.33	210 / 186
NN-Intersection (variant 1/2)	42.48 / 45.75	217 / 206
NN-KL (variant 1/2)	39.22 / 45.75	181 / 193
SMIJ [17]	33.99	-

pickup & throw (PickT) actions. The MSR2 dataset was used either split in three subsets of actions as proposed in [5] or as one dataset containing all 20 actions as in [21].

At the first experiment the 567 sequences of the database were divided into three subsets, each having 8 actions and recognition was performed separately within each subset. The separation to three difference sets was proposed by the authors in [5] due to the large computational cost of applying an algorithm in the entire database. The actions that form each subset are shown in Table 7. The AS1 and AS2 sets group actions with similar movements, while AS3 was groups complex actions together.

The overall classification rate was then computed by averaging the results obtained in each subset separately. Three different experimental setups proposed in [5] were used to verify the performance of the proposed method in this case. In the first setup (test 1), 1/3 of the sequences were used for training and the remaining 2/3 for testing. In the second setup (test 2), 2/3 of the sequences were used for training and the remaining 1/3 testing.

	HighArmW	HorizArmW	Hammer	HandCatch	FPunch	HighThrow	DrawX	DrawTick	DrawCircle	Clap	TwoHandW	Sidebox	FKick	SKick	Jog	TSwing	HighArmW
HighArmW	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
HorizArmW	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Hammer	0.00	0.00	88.89	0.00	0.00	11.11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
HandCatch	0.00	0.00	0.00	30.00	0.00	40.00	0.00	0.00	0.00	0.00	0.00	30.00	0.00	0.00	0.00	0.00	0.00
FPunch	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
HighThrow	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
DrawX	0.00	0.00	33.33	0.00	0.00	0.00	66.67	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
DrawTick	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
DrawCircle	11.11	0.00	0.00	0.00	0.00	0.00	0.00	22.22	66.67	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Clap	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
TwoHandW	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00
Sidebox	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00
FKick	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00
SKick	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00
Jog	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00
TSwing	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00
HighArmW	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00

Figure 11: Confusion matrix (17 classes) for SVM-χ^2 with variant 2 classification with $C = 206$; 90.97% overall correct classification rate. MSR1 Dataset.

Table 7: The three subsets of actions from the MSR database [5] used in the experiments as proposed in [5].

Action Set 1 (AS1)	**Action Set 2 (AS2)**	**Action Set 3 (AS3)**
Horizontal arm wave	High arm wave	High throw
Hammer	Hand catch	Forward kick
Forward punch	Draw x	Side kick
High Throw	Draw tick	Jogging
Hand clap	Draw circle	Tennis swing
Bend	Two hand wave	Tennis serve
Tennis serve	Forward kick	Golf swing
Pickup & throw	Side boxing	Pickup & throw

In the third setup (cross subject test), the sequences of half of the subjects were used for training and the rest for the testing. Tests 1 and 2 check if a

method can perform well with small/large training sets and the cross subject test checks if the method can perform well if the training and test sets consist of different subjects. The overall classification results for the three tests are shown in Table 8 along with results obtained by methods proposed in [5], [19], [20], [25] and [22].

Table 8: Overall Correct Classification rates and comparison of classification performance with the methods proposed in [5], [19], [20], [25], [22] in the three tests proposed in [5] on the MSR Action3D database.

	SVM-χ^2 (variant 1 / 2)	[5]	[19]	[20]	[25]	[22]
Test 1	88.45 / 94.5	91.6	**96.2**	95.8	96.1	93.1
Test 2	**99.12** / 97.3	94.2	97.1	97.8	97.3	95.3
Cross subject test	87.8 / **93.6**	74.7	79	82.3	91.6	92.8

As can be seen in Table 8, the proposed method surpasses all five state-of-the-art methods in Test 2 and in cross subject test by almost 2% and 1% respectively. It should be noted that the cross subject test can be considered as more fair since sequences of the same subject do not exist both in the training and in the testing set. In Test 1, the proposed method performs 1.7% worse than the best classification result achieved by [19] and surpasses the method in [5] and [22].

At the second experiment, the entire database (557 sequences) was used. The cross subject test, described above was used to assess the proposed method. The overall classification results for the cross subject test in the entire database and a comparison with state of the art methods are shown on Table 9. By observing this table one can see that the proposed method

Table 9: Correct Classification rates and comparison of classification performance in the experimental setup proposed in [21] on the MSR Action3D dataset.

	20 classes	**Number of Clusters**
SVM-χ^2 (variant 1 / 2)	88.64 / **91.94**	198 / 191
SVM-RBF (variant 1 / 2)	80.95 / 89.01	195 / 192
NN-Intersection (variant 1 / 2)	80.95 / 82.42	201 / 195
NN-KL (variant 1 / 2)	82.78 / 83.15	197 / 191
Actionlet Ensemble [21]	88.2	-
HON4D + $\mathbf{D_{disc}}$ [26]	88.89	-
Gowayyed et al. [23]	91.26	-
Barnachon et al. [24]	90.56	-
Wang et al. [28]	90.22	-
JAS (Cosine) + MaxMin [29]	83.53	-
JAS (Cosine) + MaxMin + HOG^2 [29]	**94.84**	-

(variant 2) achieved the highest classification rate when SVM-χ^2 was used for classification. The proposed method outperforms all but one of the state-of-the-art methods (JAS (Cosine) + MaxMin + HOG^2 [29]) in this challenging dataset. This is because the authors in [29] used both depth maps and skeleton information to achieve action recognition whereas the proposed method relies only on skeleton data to perform classification which is an advantage since it broadens its applicability. However, the proposed method outperforms the method proposed in [29] by almost 8.5%, when the latter takes into account only the skeleton information (JAS (Cosine) + MaxMin [29]). The confusion matrix of the best variant is shown in Fig. 12. As can be seen,

	HighArmW	HorizArmW	Hammer	HandCatch	FPunch	HighThrow	DrawX	DrawTick	DrawCircle	Clap	TwoHandW	Sidebox	Bend	FKick	SKick	Jog	TSwing	Golf	PickT	HighArmW
HighArmW	83.33	8.33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	8.33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
HorizArmW	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Hammer	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
HandCatch	0.00	16.67	0.00	75.00	0.00	8.33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FPunch	0.00	0.00	0.00	0.00	90.91	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	9.09	0.00	0.00	0.00	0.00
HighThrow	0.00	0.00	27.27	0.00	0.00	63.64	0.00	0.00	0.00	0.00	0.00	9.09	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
DrawX	0.00	0.00	7.69	0.00	0.00	7.69	76.92	0.00	7.69	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
DrawTick	0.00	0.00	0.00	0.00	0.00	0.00	0.00	93.33	6.67	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
DrawCircle	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Clap	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
TwoHandW	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Sidebox	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	80.00	0.00	13.33	0.00	0.00	6.67	0.00	0.00	0.00
Bend	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FKick	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00
SKick	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	9.09	90.91	0.00	0.00	0.00	0.00	0.00
Jog	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00
TSwing	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00
Golf	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00
PickT	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00
HighArmW	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	28.57	0.00	0.00	0.00	0.00	0.00	0.00	71.43

Figure 12: Confusion matrix (20 classes) for SVM-χ^2 with variant 2 classification with $C = 191$; 91.94% overall correct classification rate. MSR2 Dataset.

14 out of 20 actions are recognized with accuracy higher than 90%, while the action with the worst accuracy is *High Throw* (63.64%).

4.4. Futurelight dataset

The Futurelight dataset contains 155 sequences of 5 action classes in the form of bvh files, namely *dance*, *jump*, *run*, *sit* and *walk* with 30, 14, 30, 33 and 48 instances, respectively. The classes contain significant intra-class variations, making it a challenging dataset. A Leave-One-Out-Cross-Validation (LOOCV) was used to asses the performance of the method. The classification results of all variants are presented in Table 10 and are compared with the results presented in [16], [27] and [31].

As can be seen in Table 10, the best classification rate was achieved when SVM with χ^2 kernel was used for classification and the histograms were

Table 10: Correct Classification rates and comparison of classification performance in the Futurelight dataset using a leave one out cross validation.

	5 classes	**Number of Clusters**
SVM-χ^2 (variant 1/2)	96.20 / **99.37**	190 / 185
SVM-RBF (variant 1/2)	94.94 / 97.47	180 / 183
NN-Intersection (variant 1/2)	98.73 / 98.10	181 / 184
NN-KL (variant 1/2)	96.20 / 98.73	181 / 186
Ali et al. [16]	89.7	-
Raptis et al. [27]	98.03	-
Raptis et al. [31]	83.87	-

computed using variant 2 of the method. The proposed method outperforms all other state-of-the-art methods. The confusion matrix for the best variant (SVM-χ^2 with variant 2) for the Futurelight dataset using LOOCV is shown in Fig. 13.

The best classification rates in all datasets are summarized in Table 11.

Table 11: Best classification rates achieved in all datasets. C: number of clusters in K-means.

Dataset	**Classifier**	**Hist. Type**	%	**C**
MHAD	SVM-χ^2	variant 1	98.18	186
HDM05 Dataset 1 [15]	SVM-χ^2	variant 1	100	100
HDM05 Dataset 2 [17]	SVM-χ^2	variant 1	95.48	500
MSR Action3D [17]	SVM-χ^2	variant 2	90.97	206
MSR Action3D [21]	SVM-χ^2	variant 2	91.94	191
Futurelight	SVM-χ^2	variant 2	99.37	185

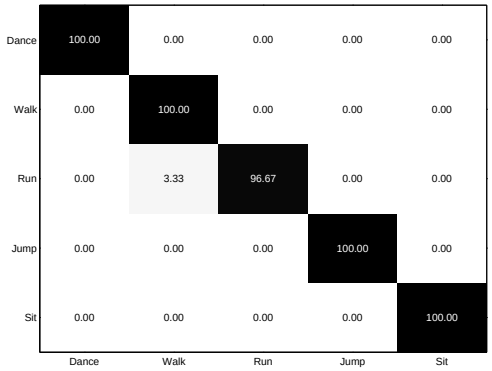


Figure 13: Confusion matrix (5 classes) for SVM-χ^2 with voting histograms classification and use of the angular K-means with $C = 185$; 99.37% overall correct classification rate.

4.5. Discussion

From the results presented in the previous subsections, it is obvious that the SVM-χ^2 classification provided the best results in all cases. This result confirms the results in [35], that χ^2 kernel provides very good results when a codebook/histogram representation is considered. The second best variant in terms of classification is the one that uses SVM with an RBF kernel (SVM-RBF). Regarding the approach used to construct the pattern histograms, variant 1 provided the best results for MHAD and HDM05 datasets and variant 2 provided the best results for MSR and Futurelight dataset. In general (see Table 11) the proposed method achieves very high recognition rates in 4 different databases, including the MSR database which was generated from Kinect data and is in general noisy. As a matter of fact, the recognition rates achieved by the method are often close to 99%. The above indicate that the proposed method performs well in a variety of (often very challenging) data, outperforming state-of-the-art methods for human action recognition

on motion capture data. The only method that performs better than the proposed one, namely [29], utilizes both mocap data and depth maps.

The proposed method achieved good classification rates for a broad range of cluster centers between 180 and 220 as can be seen in Fig. 14. In addition, in order to judge the performance of the method on a much broader range of cluster centers, the method was tested for 10, 50, 100, 500 and 1000 cluster centers. The correct classification rates of these experiments are shown in Fig. 15.

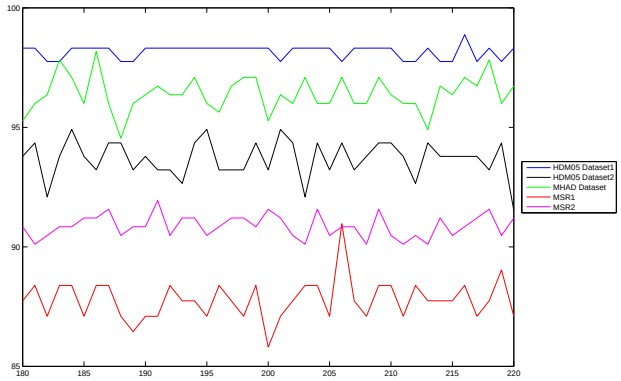


Figure 14: Classification rates for the SVM-χ^2 variant for different values of C, i.e. for different numbers of K-means clusters. Classification rates in 5 different datasets are shown

From figures 14 and 15, it is obvious that regardless the number of clusters (excluding, as expected, very small values i.e. 10), the codebooks created by the K-means algorithm can lead to vectors that describe different actions sufficiently well. Thus the proposed method does not require significant fine tuning of the parameter C (K-means clusters).

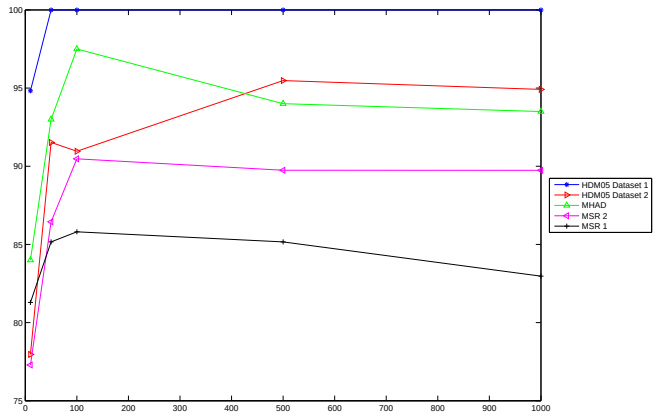


Figure 15: Classification rates for the SVM-χ^2 variant for $C = 10, 50, 100, 500, 1000$. Classification rates in 5 different datasets are shown.

5. Conclusions

In this paper, a novel method for action/movement recognition in motion capture data was proposed. The method utilizes characteristic postures (dynemes) and characteristic joint angles forward differences derived through a novel variant of the K-means algorithm, along with a bag of words approach (or a variant based on a voting scheme) and a nearest neighbour or SVM classifier. Experimental analysis verifies that the proposed approach provides very good movement recognition results surpassing all state of the art methods that rely only on motion capture data. The proposed method achieved better results mainly because it uses 2 types of features: the posture vectors (spatial features) and the forward differences (temporal/spatiotemporal features) computed over different temporal scales. The use of spatiotemporal features evaluated over different temporal scales has been proven to achieve very good activity classification results in video based-methods (see

references [36], [37] and [38]). Another reason that the proposed method provided better results is the use of χ^2 kernel, which has been shown in [35] to achieve superior results when a codebook/histogram representation is considered. A limitation of the proposed method is that it cannot distinguish between a movement and its reverse e.g. forward and backward walking. However in most applications such a distinction is not needed. In the future, extensions towards motion clustering, segmentation, indexing and retrieval, that have significant similarities with action recognition, will also be considered. Moreover, research for applying this method for action-based person identity recognition (recognizing a person based on the way he/she performs certain actions) is underway with very promising results.

References

[1] G. Burdea, P. Coiffet, Virtual Reality Technology, John Wiley & Sons, Inc., New York, NY, USA, 2 edition, 2003.

[2] R. Parent, Computer Animation, Algorithms and Techniques, Morgan Kaufmann, 2002.

[3] M. Müller, T. Röder, M. Clausen, B. Eberhardt, B. Krüger, A. Weber, Documentation Mocap Database HDM05, Technical Report CG-2007-2, Universität Bonn, 2007.

[4] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, R. Bajcsy, Berkeley mhad: A comprehensive multimodal human action database, in: Proceedings of 2013 IEEE Workshop on Applications of Computer Vision (WACV), pp. 53–60.

[5] W. Li, Z. Zhang, Z. Liu, Action recognition based on a bag of 3D points, in: Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pp. 9–14.

[6] X. Ji, H. Liu, Advances in view-invariant human motion analysis: A review, IEEE T. Syst. Man Cy. C 40 (2010) 13–24.

[7] R. Poppe, A survey on vision-based human action recognition, Image Vision Comput. 28 (2010) 976–990.

[8] P. Turaga, R. Chellappa, V. S. Subrahmanian, O. Udrea, Machine recognition of human activities: A survey, IEEE T. Circ. Syst. Vid. 18 (2008) 1473–1488.

[9] C. Li, P. R. Kulkarni, B. Prabhakaran, Segmentation and recognition of motion capture data stream by classification, Multimed. Tools Appl. 35 (2007) 55–70.

[10] J. Wang, H. Lee, Recognition of human actions using motion capture data and support vector machine, in: Proceedings of the 2009 WRI World Congress on Software Engineering, volume 1, IEEE Computer Society, Washington, DC, USA, 2009, pp. 234–238.

[11] H. Kadu, M. Kuo, C.-C. J. Kuo, Human motion classification and management based on mocap data analysis, in: Proceedings of the 2011 Joint ACM Workshop on Human Gesture and Behaviour Understanding, ACM, New York, NY, USA, 2011, pp. 73–74.

[12] S. Shariat, V. Pavlovic, Isotonic CCA for sequence alignment and activity recognition, in: Proceedings of the 2011 International Conference

on Computer Vision, ICCV '11, IEEE Computer Society, Washington, DC, USA, 2011, pp. 2572–2578.

[13] M. Raptis, D. Kirovski, H. Hoppe, Real-time classification of dance gestures from skeleton animation, in: Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, ACM, New York, NY, USA, 2011, pp. 147–156.

[14] F. Lv, R. Nevatia, M. W. Lee, 3D human action recognition using spatio-temporal motion templates, in: Proceedings of the 2005 International Conference on Computer Vision, Springer-Verlag, Berlin, Heidelberg, 2005, pp. 120–130.

[15] L. Deng, H. Leung, N. Gu, Y. Yang, Generalized model-based human motion recognition with body partition index maps, Comput. Graph. Forum 31 (2012) 202–215.

[16] S. Ali, A. Basharat, M. Shah, Chaotic invariants for human action recognition, in: Proceedings of IEEE 11th International Conference on Computer Vision, 2007. ICCV 2007., pp. 1–8.

[17] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, R. Bajcsy, Sequence of the most informative joints (smij): A new representation for human skeletal action recognition, J. Vis. Commun. Image R. In Press (2013) –.

[18] L. Han, X. Wu, W. Liang, G. Hou, Y. Jia, Discriminative human action recognition in the learned hierarchical manifold space, Image Vision Comput. 28 (2010) 836–849.

[19] L. Xia, C.-C. Chen, J. Aggarwal, View invariant human action recognition using histograms of 3D joints, in: Proceedings of the 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)., pp. 20–27.

[20] X. Yang, Y. Tian, Eigenjoints-based action recognition using naïve-bayes-nearest-neighbor, in: CVPR Workshops, IEEE, 2012, pp. 14–19.

[21] J. Wang, Z. Liu, Y. Wu, J. Yuan, Mining actionlet ensemble for action recognition with depth cameras, in: Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1290–1297.

[22] M. Devanne, H. Wannous, S. Berretti, P. Pala, M. Daoudi, A. Bimbo, Space-time pose representation for 3D human action recognition, in: A. Petrosino, L. Maddalena, P. Pala (Eds.), New Trends in Image Analysis and Processing ICIAP 2013, volume 8158 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2013, pp. 456–464.

[23] M. A. Gowayyed, M. Torki, M. E. Hussein, M. El-Saban, Histogram of oriented displacements (HOD): Describing trajectories of human joints for action recognition., in: Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI/AAAI, 2013.

[24] M. Barnachon, S. Bouakaz, B. Boufama, E. Guillou, Ongoing human action recognition with motion capture, Pattern Recognition 47 (2014) 238 – 247.

[25] X. Yang, C. Zhang, Y. Tian, Recognizing actions using depth motion maps-based histograms of oriented gradients, in: Proceedings of the 20th ACM international conference on Multimedia, MM '12, ACM, New York, NY, USA, 2012, pp. 1057–1060.

[26] O. Oreifej, Z. Liu, Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences, in: Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition, CVPR '13, IEEE Computer Society, Washington, DC, USA, 2013, pp. 716–723.

[27] M. Raptis, K. Wnuk, S. Soatto, Flexible Dictionaries for Action Classification, in: Proceedings of the 1st International Workshop on Machine Learning for Vision-based Motion Analysis - MLVMA'08, Marseille, France.

[28] C. Wang, Y. Wang, A. L. Yuille, An approach to pose-based action recognition, in: Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)., IEEE Press, 2013, pp. 915–922.

[29] E. Ohn-Bar, M. M. Trivedi, Joint angles similiarities and HOG^2 for action recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops: Human Activity Understanding from 3D Data, CVPR '13, IEEE PRESS, 2013.

[30] J. Barbič, A. Safonova, J.-Y. Pan, C. Faloutsos, J. K. Hodgins, N. S. Pollard, Segmenting motion capture data into distinct behaviors, in: Proceedings of the Graphics Interface 2004, GI '04, Canadian Human-

Computer Communications Society, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2004, pp. 185–194.

[31] M. Raptis, K. Wnuk, S. Soatto, Spike train driven dynamical models for human actions., in: Proceedings of the Computer Vision and Pattern Recognition, IEEE, 2010, pp. 2077–2084.

[32] F. Zhou, F. Torre, J. Hodgins, Aligned cluster analysis for temporal segmentation of human motion, in: Proceedings of the 8th IEEE International Conference on Automatic Face Gesture Recognition, 2008. FG '08., pp. 1–7.

[33] N.Nikolaidis, I.Pitas, Nonlinear processing and analysis of angular signals, IEEE T. Signal Proces. 46 (1998) 3181 – 3194.

[34] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE T. Pattern Anal. 22 (2000) 888–905.

[35] J. Zhang, M. Marszałek, S. Lazebnik, C. Schmid, Local features and kernels for classification of texture and object categories: A comprehensive study, Int. J. Comput. Vision 73 (2007) 213–238.

[36] F.-F. Li, P. Perona, A bayesian hierarchical model for learning natural scene categories, in: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2 - Volume 02, CVPR '05, IEEE Computer Society, Washington, DC, USA, 2005, pp. 524–531.

[37] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: Spatial

pyramid matching for recognizing natural scene categories, in: Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, volume 2, pp. 2169–2178.

[38] J. Choi, W. J. Jeon, S.-C. Lee, Spatio-temporal pyramid matching for sports videos, in: Proceedings of the 1st ACM International Conference on Multimedia Information Retrieval, MIR '08, ACM, New York, NY, USA, 2008, pp. 291–297.